Online Change Estimation Models for Dynamic Web Resources*

A Case-Study of RSS Feed Refresh Strategies

Roxana Horincar, Bernd Amann, and Thierry Artières

LIP6 - University Pierre et Marie Curie, Paris, France
{roxana.horincar,bernd.amann,thierry.artieres}@lip6.fr

Abstract. Modern web 2.0 applications have transformed the Internet into an interactive, dynamic and alive information space. Personal weblogs, commercial web sites, news portals and social media applications generate highly dynamic information streams which have to be propagated to millions of users. This article focuses on the problem of estimating the publication frequency of highly dynamic web resources. We illustrate the importance of developing efficient *online* estimation techniques for improving the refresh strategies of RSS feed aggregators like Google Reader [8], Datasift [7] or Roses [11]. We study the temporal publication characteristics of a large collection of real world RSS feeds and we define and evaluate several online estimation methods in cohesion with different refresh strategies. We show the benefit of using periodical source publication patterns for change estimation and we highlight the challenges imposed by the application context.

1 Introduction

Understanding how web resources evolve in time is important for conceiving tools designed to ease the interaction between people and dynamic web content published by online newspapers, commercial web sites, social networks and collaborative web sites like Wikipedia. Most of these information sources can only be accessed via standard pull-based web protocols (HTTP) and estimating the degree of information change during a given time period is crucial for developing efficient refresh strategies.

Modern web sites, such as online newspapers or social media sites, publish their stream of changes in form of light-weight RSS/Atom feeds for reducing the communication cost between servers and clients. Technically speaking, an RSS feed is a standard XML document containing a list of time-stamped text descriptions including links to the corresponding web pages. The size of this list is generally limited to a constant value, where the publication of a new item usually removes the oldest one in the corresponding window. From the user's point of view, RSS documents are perceived as a stream of items pushed to their screen.

* The authors acknowledge the support of the French Agence Nationale de la Recherche (ANR), under grant CARTEC (ANR-07-MDCO-016)

M. Brambilla, T. Tokuda, and R. Tolksdorf (Eds.): ICWE 2012, LNCS 7387, pp. 395–410, 2012.

However, when considering the underlying communication protocol, there is no distinction between RSS feeds and other web resources. Both kinds of resources have to be refreshed by using the standard pull-based HTTP protocol where changes can only be detected by explicitely contacting the server.

As any web resource, RSS feeds evolve independently of their clients which must take their refresh decisions by estimating the change frequencies [1, 17]. In this paper, we focus on the problem of estimating the change frequency of dynamic web data. Our first goal is to improve the refresh strategies of RSS aggregators, but other web data processing systems like web crawlers or web data warehouses may as well benefit from the techniques presented in this article. The main challenges we address are:

Rapidly changing publication behavior : Event-related feed sources like topic based news feeds or social media feeds (Twitter) may suddenly change their publication frequency related to a particular event (e.g. twitter hash-tag). This data dynamics leads to the necessity of continually updating the publication frequency estimation, using online estimation techniques.

Incomplete knowledge : Another challenge is the limited access bandwidth due to standard web politeness policies and limited computing, network and storage resources. Estimators then have to deal with incomplete knowledge about the data change history, not knowing how often, how much and when exactly a source produces new information items.

Irregular estimation intervals : In many web applications, data sources are not refreshed in regular time intervals. The exact access moment is generally decided by a refresh strategy, usually conceived to optimize certain quality measures within a minimum cost. Irregular refresh periods also make the estimation process more challenging.

Our main contributions are:

- an analysis of general characteristics with a focus on the temporal dimension of real RSS feed sources using data collected over four weeks from more than 2500 RSS feeds,
- two online estimation methods that correspond to different RSS publication activity models and
- an experimental evaluation of the online estimation methods in cohesion with different refresh strategies and an analysis of their effectiveness on sources with different publication behavior.

The rest of this paper is organized as follows. Section 2 gives a short survey of related work on refresh strategies and parameter estimation for web data. In section 3 we describe the problem and benefits of online change estimation in the context of web data refresh strategies. Section 4 proposes two ways to model the publication activity of a source and introduces some methods for updating these publication models online. Section 5 analysis the temporal characteristics of two collections of real RSS feed sources. Section 6 exposes the experimental results obtained by evaluating the proposed online estimation methods in conjunction with different refresh strategies. Conclusions and future work are presented in section 7.

2 Related Work

The problem of efficiently refreshing dynamic web information is largely studied in the context of web pages [4–6, 12–14] and RSS feed [11, 16, 17]. The majority of these strategies are based on the widely accepted assumption that web resources follow a Poisson process [15] characterized by a change rate parameter $\lambda(t)$ which can be estimated by observing the change history of a web page.

Considering $\lambda(t) = \lambda$ to be constant corresponds to a *homogeneous* (opposed to *non-homogeneous*) Poisson process which represents a stateless and time-independent random process where events occur with the same probability (rate) λ at every time point. It has been shown that this model is appropriate for a time granularity of at least one month [4–6]. On the other hand, for time granularities shorter than a month, researchers have shown that the homogeneous Poisson model is no longer suited [2, 9].

Offline refresh strategies [4, 5, 16, 17] assume that the change frequency of web pages or posting rates of web streams is known a-priori. They usually use average values measured beforehand or learnt during an initial learning phase with access to a complete changing history. This assumption is sufficient for a low frequency refresh activities where each web resource is refreshed rarely (like in web search engines) and the update frequency can be averaged over long time periods [4, 5, 16, 17].

Reference [6] presents several change (frequency) estimators for web pages, assuming an incomplete change history with irregular refresh frequencies. They show that a Web crawler could achieve 35% improvement in "freshness" simply by adopting their estimator. However, their analysis is based on the hypothesis that the date of the last change or the existence of a change on a web page are known in advance for estimation.

Based on the previous observations, [16] uses a periodic (inhomogeneous) Poisson model with a daily periodicity within a RSS feeds scenario. Similarly, [12] presents an empirical study of two *online refresh strategies* that use a curve-fitting over a generative model method and conservative bounds to dynamically adjust refresh parameters.

In the context of information filtering (also referred to as publish/subscribe), a user subscribes to the system to receive notifications whenever certain events of interest take place (e.g., when a document that corresponds to a certain filtering condition becomes available). In order to estimate the probability that a node has published new information relevant to a user's subscription, [18, 19] use time series prediction techniques for approximate information filtering. Our work uses a similar approach in a different context.

3 Refresh Strategies and Online Change Estimation

Large-scale web applications like web search engines, web archives, web data warehouses, publish-subscribe systems and news aggregators have to collect information from a large number of dynamic web resources. In order to accomplish

this task efficiently, these systems are generally based on *refresh strategies* for deciding when to refresh each source in order to maximize one or several quality criteria under limited resources.

Refresh decisions are based on appropriate source *publication models* for making predictions. There exist various publication models. Content-independent models [6] estimate the probability that a source has changed at least once or n times at some time instant t, whereas content-dependent models [12] might include some heuristics for estimating the importance of change between two versions. We consider in this article the case of a RSS aggregator node which is subscribed to a collection of sources. Let t_0 represent the last time instant when source s has been refreshed by the aggregator. We define a *divergence* function $Div(s, t, t_0)$ estimating the total number of new items published by the source s in the time period $(t_0, t]$. Obviously the quality (preciseness) of this estimation is important for the quality of the corresponding refresh strategy [4, 5].

A traditional way for estimating divergence is to define the behavior of a source s as a stochastic (Poisson) process which can be characterized by time dependent *publication frequency* variable $\lambda(s, t)$ that measures the number of items published by source s at time instant t. Divergence can then be defined as an integral of publication frequency $\lambda(s, t)$ over time:

$$Div(s, t, t_0) = \int_{t_0}^{t} \lambda(s, x)dx \tag{1}$$

In practice, refresh strategies use a discrete time dimension, where time periods are divided into time units of fixed size and divergence is defined by a sum of divergence estimations for the intervals (see section 4).

Online and offline change estimation: The general refreshing process illustrated in figure 1 is accomplished by (1) the *refresh strategy* which *uses* the publication model for estimating the divergence and the next refreshing time moment of each source and (2) the *change estimator* which generates and updates the publication model. In an *offline scenario* the change estimator module does not exist. The refresh strategy uses a precomputed publication model which is updated offline (independently of the refresh process). *Online estimation* interleaves both tasks and each new observation (obtained by a refresh) is used immediately for updating the publication model.

Why online change estimation is important: Keeping the estimated publication frequency of a source constant over a long period of time can represent an important source of errors if the source publication activity changes in time.

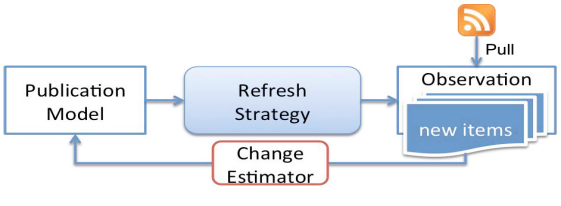


Fig. 1. Online estimation

This is illustrated in figure 2 showing the evolution of the real and the estimated divergence of a source during a day. Figure 2 compares (for a given source) the real divergence values (red curve) with the estimated values using a constant publication frequency (offline estimated divergence as the green curve) and the estimated values using an adaptive publication frequency (online estimated divergence as the blue curve). In both curves, the source is refreshed in regular time intervals which resets the divergence values to 0. The green estimated divergence function presented in figure 2 increases with a constant slope because it is based on a constant publication frequency (previously learnt in an offline manner and not updated afterwards). Differently from this case, the blue estimated divergence function in figure 2 is computed based on a publication frequency that continuously adapts its value in time (online estimation), converging to a zero publication frequency when the source does not publish anything and increasing as the source starts publishing. The estimation obviously is better in average in the second case.

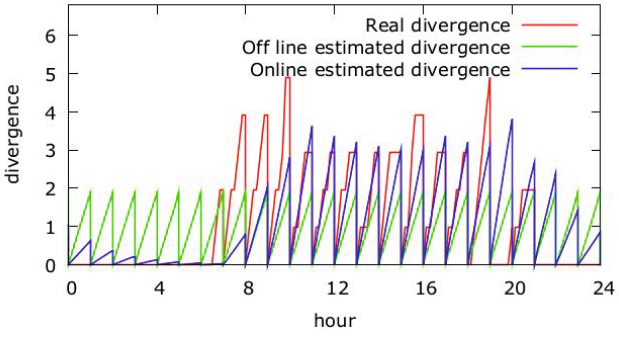


Fig. 2. Real vs. estimated divergence

4 Online Change Estimation for RSS Feeds

Our approach for estimating the change rate of RSS feeds is strongly inspired from standard results in time series analysis [3]. These techniques are used to predict future time series values based on past observations and are usually based on the hypothesis that both observations and predictions are done at equally spaced time intervals. In our particular case, the observations are made at the moment of a refresh, which is decided by the refresh strategy used by the crawler [11]. This makes the prediction process less precise than in the case of classical time series model usage.

We base our online estimation methods on observations of the number of occurred changes, i.e. new items published by a feed. In the particular case of working with RSS feeds, we could have chosen to use the specific RSS field $< pubDate >$ in order to find out exactly the publication date of each item. Nevertheless, we prefer to ignore this attribute for two reasons. First, [10] reports that this information ($< pubDate >$) is missing in about 20% of items. Second,

ignoring this particular kind of metadata keeps our estimation methods generic and adaptable for other kinds of data (e.g. web pages).

4.1 Single Variable Publication Model

Estimating Divergence: Our first publication model represents the publication frequency of a source s at time t by a single variable, $\lambda(s,t)$. Let T_r represent the time instant of the r^{th} refresh of s and $\lambda^r = \lambda(s,T_r)$ be the change rate of source s estimated at time instant T_r. Then the divergence of s at time instant $T \in [T_r, T_{r+1})$ can be simply estimated by the following formula:

$$Div^{est}(s,T,T_r) = (T - T_r) \cdot \lambda^r$$

Updating Frequency Estimation: Let $x^{r+1} = Div(s,T_{r+1},T_r)$ be the number of new items published since the last refresh at T_r and observed at T_{r+1}. The newly estimated value of the publication frequency is obtained by single-exponentially smoothing the new observation with the previous estimation:

$$\lambda^{r+1} = \alpha \cdot \frac{x^{r+1}}{(T_{r+1} - T_r)} + (1-\alpha) \cdot \lambda^r$$

This estimation method relies on all previous observations, with exponentially decaying weights, parameter $\alpha \in [0,1]$ representing the smoothing constant.

4.2 Periodic Publication Model

Our second estimation model of publication is based on the hypothesis of periodicity. In this case, the publication frequency of a source is described as a periodic function with some (constant) period Δ_T: $\lambda(s,t) = \lambda(s,t+\Delta_T)$.

We use a *discrete* representation of the publication frequency as a table $P(s)$ of n values, each corresponding to a time slot $[t_i, t_{i+1})$, $i \in \{0,...n-1\}$. Each time slot is of constant size $t_{i+1} - t_i = \Delta_T/n$. We will call $P(s)$ the publication model of s. Then $\lambda_i(s,t)$ corresponds to the $(i+1)^{th}$ value in $P(s)$ where $(t \mod \Delta_T) \in [t_i, t_{i+1})$ (i is the time slot covering t). In the following we denote by λ_i the average publication rate of source s during time slot i. In our experiments (section 6) we use a daily publication model where $\Delta_T = 24$ hours, $n = 24$ time slots of 1 hour each.

Estimating Divergence: Let T_r represent the time instant of the r^{th} refresh of s and $P_r(s) = \{\lambda_i^r\}$, $i \in \{0,...n-1\}$ be the publication model of s estimated at time instant T_r. Then the expected divergence of s at time instant $T \in [T_r, T_{r+1})$ can be estimated by the following formula where i corresponds to the time slot containing T_r and there are $k+1 = (\lceil T \rceil - \lfloor T_r \rfloor) \cdot n/\Delta_T$ time slots "covered" by the interval $[T_r, T)$ (the definitions are illustrated in figure 3):

$$Div^{est}(s,T,T_r) = \int_{T_r}^{T} \lambda_j^r(s,t)\, dt =$$

$$= \lambda_i^r(t_{i+1} - T_r) + \Delta_T/n \cdot \sum_{j=i+1}^{i+k-1} \lambda^r_{(j \mod n)} + \lambda^r_{i+k}(T - t_{i+k})$$

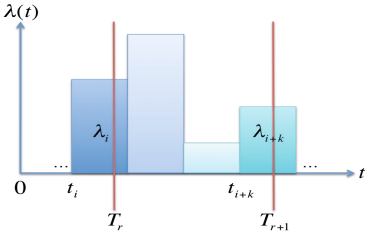


Fig. 3. Periodic publication model

Updating Frequency Estimation: Suppose that the aggregator refreshed some source s at some time moments T_r and T_{r+1} that correspond to time slots i and $i+k$. At T_{r+1}, the aggregator fetches $x^{r+1} = Div(s, T_{r+1}, T_r)$ new items published since the last refresh at T_r. The intuitive idea of the model update is to distribute the last observed items x^{r+1} in the time interval $[T_r, T_{r+1})$. This distribution is done *proportionally* to the expected divergence $Div^{est}_{r+1} = Div^{est}(s, T_{r+1}, T_r)$ estimated using the values of λ^r_j that correspond to the time interval $[T_r, T_{r+1})$. We compute λ^{r+1}_j as the newly predicted value of λ_j that corresponds to time slot j as follows:

$$\lambda_j^{r+1} = \begin{cases} \alpha \cdot \frac{\lambda_j^r}{Div^{est}_{r+1}} \cdot x^{r+1} + (1-\alpha) \cdot \lambda_j^r & \textbf{if } j \in \{i, ...i+k\} \\ \lambda_j^r & \textbf{otherwise} \end{cases}$$

where $\alpha \in [0, 1]$ represents a smoothing parameter that is used to give more or less weight to recent observations. This reestimation formula corresponds to a maximum likelihood estimate of the publication frequencies λ_j based on the observation x^{r+1} at iteration $r+1$, smoothed with the estimates at previous iteration r.

5 Dataset Description

In order to better understand the change estimation problem, we studied a collection of real world RSS feeds focusing on their temporal dimension. We used two different datasets: dataset 1 was obtained from crawling a list of feeds [10] harvested from major RSS directories, portals and search engines (such as syndic8.com, Google Reader, feedmil.com, completeRSS.com etc.) and dataset 2 was acquired from a manually chosen list of RSS news feeds of different online newspaper websites, both French (such as Le Monde, Le Figaro, AFP) and international (such as CNN, New York Times, Euro News). We selected 1658 RSS crawled feeds from dataset 1 and 963 RSS news feeds from dataset 2 that had at least one posting within the four-week period between 14 March - 10 April 2011.

Publication Activity: In figure 4 we show the distribution of feeds for various activity classes defined by different posting rates for the two different datasets. The distributions show that feeds with very slow publication activity are predominant, while roughly 20% of the feeds publish more than 10 items daily.

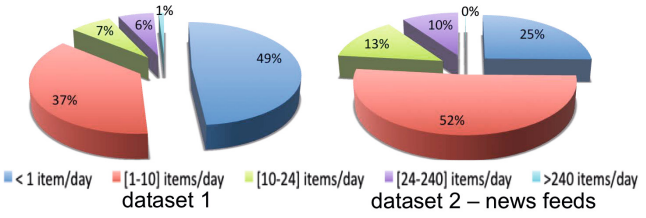


Fig. 4. Feeds per activity class

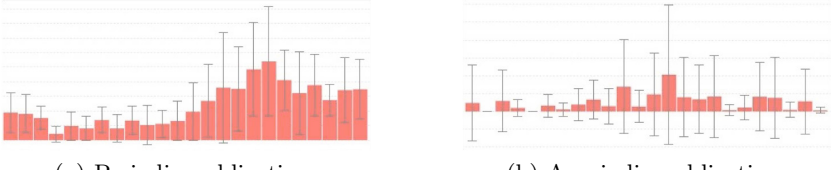
It has been shown in [10] that whereas the number of productive feeds is quite small, they are the ones that produce most of the items: 17% of RSS/Atom feeds produce 97% of the items.

Feed Periodicity: It is widely accepted that the past change represents a good predictor of future change. This works well especially for those types of feeds that have a foreseeable publication activity, for example, feeds that publish daily the same number of items. In this sense, measurements on real data done in [16] show that most of the daily posting rates of feed sources are stable, at least for their dataset, within the 3-month period they used for their experiments. But there are also feeds whose publication behavior vary in time, both in the number of daily published items and in the shape of publication activity.

In order to detect changes in publication frequency, for each hour (time slot i) of a day, we logged the number of items published by a feed and then computed the *mean* μ_i and the *standard deviation* σ_i on the entire period. We consider that a small *coefficient of variation* CV value is representative for periodic feed sources.

$$CV = \frac{1}{24} \sum_{i=0}^{23} \frac{\sigma_i}{\mu_i} \qquad \text{where } \mu_i \neq 0$$

When the mean values are close to zero, the coefficient of variation becomes sensitive to small changes in the means and inappropriate for testing sources with a low publication activity. Testing for $CV \leq 1$, we discovered that periodic sources represent 20% of the sources in our datasets that publish more than 10 items per day and 50% of the sources that publish more than 48 items per day. As an example, in figures 5a and 5b we represented the average (pink bars) and the standard

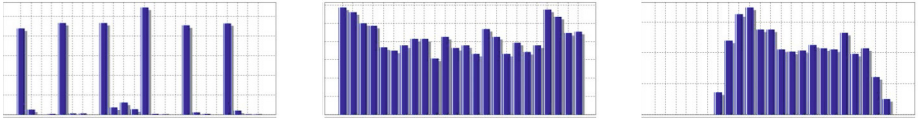


(a) Periodic publication (b) Aperiodic publication

Fig. 5. Periodic and aperiodic publication behavior

deviation (vertical lines) of the number of published items at different time slots, one for each hour of the day, for a periodic and an aperiodic feed.

Publication Shape: We also studied the feed collection looking for different "shapes" in the daily publication activity. The shape of a daily publication model highly depends on what happens "behind the curtains" of each feed. Some feeds may be generated by human activity, while others may be based on some automatic publication process. We classified the feeds in three different categories, as shown in figure 6: feeds that have peaks, usually generated by an automatic publication robot, that have a uniform publication activity, such as in the case of a news aggregator and those that exhibit waves, following the regular daily schedule of a human activity. This classification has been obtained by using a shape discovery heuristic that uses two thresholds, inferior and superior to the average number of items published during an hour, to distinguish between hours with insignificant, average or very high publication activity.



(a) Peaks (b) Uniform (c) Waves

Fig. 6. Publication shapes: peaks, uniform and waves

In figure 7 we show the distribution of feeds for various activity classes and publication shapes for dataset 1 (similar results were obtained for dataset 2). The distributions show that feeds with very slow publication activity tend to publish more with peaks, the uniform pattern is very much present in feeds with very high publication activity while the wave shape appears in feeds with low, medium and high publication frequencies.

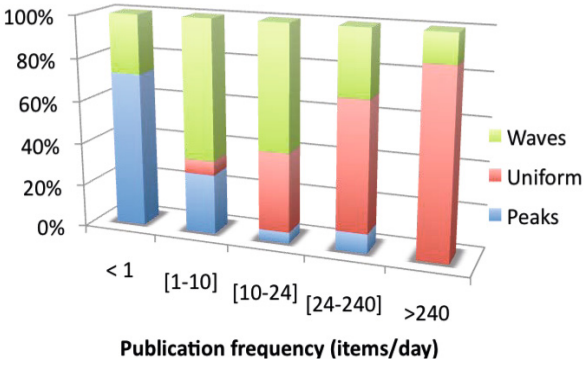


Fig. 7. Feeds per publication shape and activity class - dataset 1

6 Experimental Evaluation

In this section, we evaluate the performance of our online estimation methods in cohesion with different refresh strategies based on real RSS feeds data collected during a four-week period (see section 5).

Setup: We focused our interest on feeds with a relatively high publication activity. For our experiments, we selected (using the shape discovery heuristic) three subsets of 10 feed sources each, representative for the three publication shapes, having a publishing activity of at least 10 items per day.

We emulated the source publication activity by constructing a cycle-based environment, where a cycle corresponds to a time unit of duration 10 minutes. Furthermore, we worked with a normalized source publication, i.e. instead of publishing x items during a time slot, we consider that a source publishes x/N items, where N represents the total number of items published by the source during that entire day. Working this way, we focused ourselves on estimating the shape of a source publication activity and we avoided the influence of any strong fluctuation in terms of total number of items published daily.

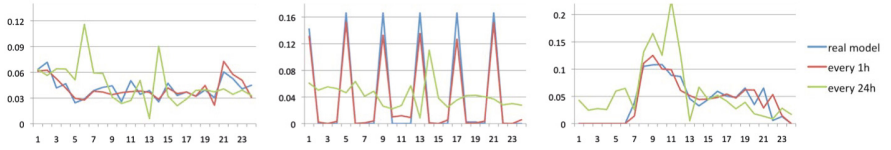
Choosing the optimal value of the smoothing parameter α depends on the type of the source, on the refresh frequency and on the level of convergence of the source publication model. In each case, we chose an experimentally found value of α such that it minimizes the divergence errors, usually using values in the interval $[0.01 - 0.2]$.

6.1 Online Estimation Evaluation

In order to evaluate the online estimation techniques presented in section 4, we applied an *uniformly distributed random* refresh strategy, in which the refreshes are done at irregular intervals of time that are uniformly distributed around a fixed average value. For example, when we say that a source is refreshed on average every 1 hour, that means that it can be refreshed within the interval 10 minutes - 2 hours. We put all sources in the same initial conditions, initializing their publication models at 0 and started the evaluation after an initial warm up period.

Robustness of the Periodic Publication Estimation: In order to test the robustness of our periodic publication estimation, how it acts to sudden changes in the publication behavior of the sources and how it is influenced by the refresh frequency used by the strategy, we created an artificial source. We concatenated publication activities from three sources with different types of publication shapes: 16 weeks of uniform, followed by 16 weeks of peaks and followed by 16 weeks of waves.

Experiments were done using the *uniformly distributed random* strategy that refreshed the source every 1 hour and every 24 hours on average. We logged the estimated daily publication models at the end of each week. We also defined the "real" daily publication model as an average done on the 7 days of source publication activity previous to the measurement moment. In figure 8 we present



(a) After 16 weeks: Uni-form (b) After 32 weeks: Peaks (c) After 48 weeks: Waves

Fig. 8. Daily publication model: real vs. estimated model

in detail the real and estimated daily publication models of the artificial source just before each change in the publication behavior, i.e. at the end of 16th, 32nd and 48th week (time moments circled and marked with vertical blue lines in figure 9). Furthermore, we compute the 24-dimensional Euclidean (2-norm) distance between the real and the estimated daily publication models after each week and present it in figure 9.

Experiments shown in figures 8 and 9 prove the bad influence a small refresh frequency can have on the quality of the estimation process. Convergence speed of the publication estimations are shown in figure 9: while the estimated daily publication model obtained with a refresh done every 1 hour on average converges rapidly towards the real model, the estimated model obtained with a refresh done every 24 hours oscillates and diverges in time.

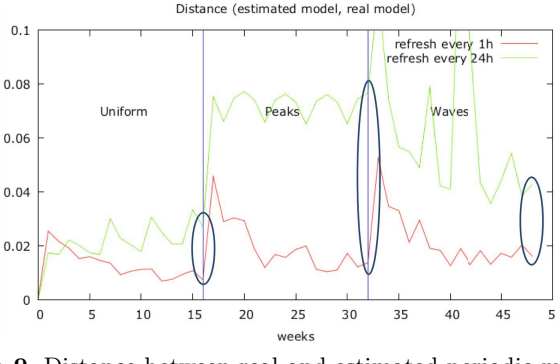
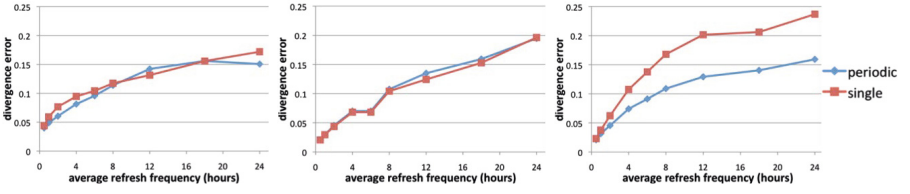


Fig. 9. Distance between real and estimated periodic model

Online Estimation Quality: At each cycle t, we computed the root mean squared error of the estimated divergence (defined in section 3) for all sources $s_i \in S$, separately for the periodic and for the single variable publication model, as follows:

$$divErr = \sqrt{\frac{1}{|S|} \cdot \sum_{s_i \in S} (Div(s_i, t, t_0)^{real} - Div(s_i, t, t_0)^{est})^2} \qquad (2)$$

Results are presented in figure 10, separately for the three types of sources with different publication shapes: peaks, uniform and waves. Each point represents the average of the root mean squared divergence errors computed during the simulation, that were obtained for different refresh frequencies. The values used for the refresh frequencies are shown in hours and they range from a refresh done every 30 minutes to every 24 hours on average.



(a) Peaks (b) Uniform (c) Waves

Fig. 10. Divergence error

Experiments show clearly that in the case of waves, the periodic estimation obtains better results than the single variable one in terms of minimal divergence error. Since it is more precise, it estimates better the wavy source publication behavior, no matter how often the sources are refreshed and thus, how often the publication model is updated. In the case of peaks, the difference between the two publication estimations is less striking. When the sources are refreshed often and therefore the learnt periodic publication model is precise, the periodic estimation obtains smaller divergence errors. As the sources are refreshed less frequently, the single variable estimation becomes as good as the periodic one; this happens for two reasons: first, the periodic model becomes less accurate and thus it diminishes its performance and second, our feed sources exhibit their peaks at very regular intervals, e.g. every 4 hours, as shown in figure 6a, and this advantages the single variable publication model for refresh frequencies larger than the average interval in between peaks. As for the uniform sources, both single and periodic publication estimations perform similarly, with the observation that the single variable publication model should be preferred because it is much more simple to use and update. The feeds concerned by this case, that publish in a uniform manner, represent 57% of the feeds with high publication rate (more than 1 item published per hour), as we observed on our real feeds datasets (section 5).

6.2 Integration of Online Estimation with 2Steps Refresh Strategy

We also integrated and tested the cohesion between our online estimation techniques with the optimal 2*steps* refresh strategy introduced in [11], whose efficient results highly depend on the quality of the used publication models.

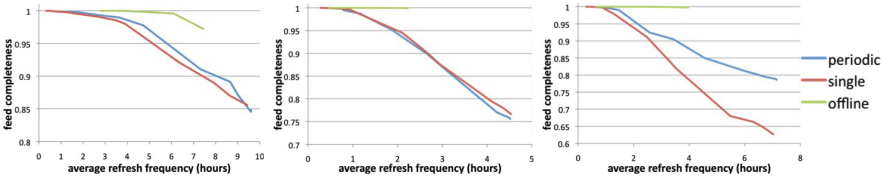
In order to better understand the following, we briefly introduce some further notions. A RSS feed is represented by a limited number of items available at some

time instant, called a *publication window* of size W_s. We call a source *saturated* if the total number of new items published since its last refresh time reaches the capacity of the publication window W_s. After the saturation point, if the source is still not refreshed, the aggregator node starts to lose items, since the arrival of new items will replace items that have not been read yet by the aggregator.

It is important to mention that we ignored the saturation problem when updating the publication models, but for the evaluation of the 2*steps* refresh strategy we considered that sources have a publication window of $W_s = 20$ items. We chose to do that in order to help the online estimation by giving it unbiased information as input, but one must be aware that saturation can not be avoided in real world RSS feed aggregation systems.

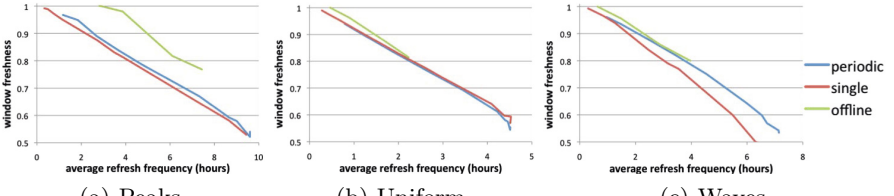
As before, we evaluate the online estimation quality by measuring the divergence error (equation 2). The results obtained for the sources having different publication shapes are similar with those obtained when testing with the uniformly distributed refresh strategy (see figure 10).

Furthermore, we test the effectiveness of the 2*steps* refresh strategy in terms of feed completeness and window freshness (quality measures defined in [11]), in the cases where the strategy uses *offline* information on the publication model of the sources and publication models estimated with the online estimation techniques presented here (*periodic* and *single variable* publication estimation). The results obtained for the feed completeness and window freshness are presented in figures 11 and 12.



(a) Peaks (b) Uniform (c) Waves

Fig. 11. Feed Completeness



(a) Peaks (b) Uniform (c) Waves

Fig. 12. Window Freshness

When sources are refreshed very frequently (big bandwidth), both periodic and single variable publication estimation give very good results in terms of feed completeness and window freshness, no matter the source publication shapes. Frequent refreshes alone assure high scores for quality measures and besides that, good convergence for both periodic and single variable publication models. In the case of peaks, when the aggregator refreshes rarely, sources become saturated very often and the $2steps$ strategy focuses itself on refreshing those saturated ones. Predicting when a source publishes $W_s = 20$ items in the case of sources with regular peaks works well both with the periodic and the single variable publication model, because in this case the precision offered by the periodic model (that knows exactly at which point in time each item was published) is useless. All these make that both periodic and single variable publication estimation give similar results in terms of feed completeness and window freshness for the peaks in case of rare refreshes. When sources are refreshed more often and there are less saturated sources, periodic publication estimation give better results. In the case of wavy publication behavior, periodic estimation outperforms the single variable one because of the information accuracy it provides, no matter how often the sources are refreshed. In this case it is the most clear how the preciseness of the information on which a refresh strategy is based influences its performances. For the uniform sources, the same conclusion as for the uniformly distributed random strategy holds. Because results are similar and especially because the single variable publication model is far more easy to use and update, this last one should be used.

6.3 Discussion

Experimental results illustrate the high cohesion between the correctness of the decisions made by a refresh strategy and the publication model used together with the quality of the estimation process. It has been shown that the refresh frequency used by the strategy has an important influence on the quality of the estimation process. Furthermore, saturation has a highly negative impact: if refreshes are not done often enough and items are lost, the estimation process uses inaccurate data for updating the model. In this case, a possible solution is the separation of the estimation from the refresh process of the crawling module, thus separating the bandwidth resources needed for the two processes.

When the refresh strategy has strong constraints in terms of bandwidth usage, online estimation does not represent a reliable solution. One alternative solution is then to allocate separate bandwidth for learning a publication profile (offline scenario) and then to use the precomputed model to refresh the sources, without updating it. This gives good results for feeds (or queries on feeds) that do not change their publication behavior in time, but it is not advisable to be used for specific queries that are very dynamic. Moreover, several such learning periods may be repeated to update periodically the source publication profiles. Since a refresh strategy is based on a publication model and the estimation of the publication model depends on the bandwidth allocated by the refresh strategy, finding the optimal balance between the two represents a challenge.

7 Conclusion

In this paper we have investigated problems related to an RSS aggregator that retrieves information from multiple RSS feed sources automatically. In particular, we have proposed and studied two online estimation methods that correspond to two different models of the source publication activity. We tested the online estimation methods in cohesion with different refresh strategies. We compared these methods for different publication activity shapes and we highlighted the challenges imposed by the application context. In addition, we studied the characteristics of real world RSS feeds datasets focusing on the temporal dimension.

We consider several directions for future work. First, we plan to add other learning components for estimating the total number of items published during a day. Also, we want to integrate an algorithm that adjusts dynamically the value of the smoothing parameter α to the optimal value that assures minimal estimation errors. Finally, for reducing estimation cost, we intend to introduce clustering techniques for grouping source feeds with similar publication activities.

References

1. Adam, G., Bouras, C., Poulopoulos, V.: Utilizing RSS Feeds for Crawling the Web. In: 2009 Fourth International Conference on Internet and Web Applications and Services, pp. 211–216. IEEE (2009)
2. Brewington, B.E., Cybenko, G.: How dynamic is the web? Computer Networks 33(1-6), 257–276 (2000)
3. Chatfield, C.: The Analysis of Time Series: An Introduction. CRC Press (2004)
4. Cho, J., Garcia-Molina, H.: Synchronizing a database to improve freshness. SIGMOD Rec. 29(2), 117–128 (2000)
5. Cho, J., Garcia-Molina, H.: Effective page refresh policies for web crawlers. ACM Trans. Database Syst. 28(4), 390–426 (2003)
6. Cho, J., Garcia-Molina, H.: Estimating frequency of change. ACM Trans. Internet Technol. 3(3), 256–290 (2003)
7. Datasift, `http://datasift.com/`
8. Google reader, `http://www.google.com/reader`
9. Gruhl, D., Guha, R.V., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. In: Feldman, S.I., Uretsky, M., Najork, M., Wills, C.E. (eds.) WWW, pp. 491–501. ACM (2004)
10. Hmedeh, Z., Vouzoukidou, N., Travers, N., Christophides, V., du Mouza, C., Scholl, M.: Characterizing Web Syndication Behavior and Content. In: Bouguettaya, A., Hauswirth, M., Liu, L. (eds.) WISE 2011. LNCS, vol. 6997, pp. 29–42. Springer, Heidelberg (2011)
11. Horincar, R., Amann, B., Artières, T.: Best-Effort Refresh Strategies for Content-Based RSS Feed Aggregation. In: Chen, L., Triantafillou, P., Suel, T. (eds.) WISE 2010. LNCS, vol. 6488, pp. 262–270. Springer, Heidelberg (2010)
12. Olston, C., Pandey, S.: Recrawl scheduling based on information longevity. In: WWW 2008: Proceeding of the 17th International Conference on World Wide Web, pp. 437–446. ACM, New York (2008)
13. Olston, C., Widom, J.: Best-effort cache synchronization with source cooperation. In: SIGMOD 2002: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, pp. 73–84. ACM, New York (2002)

14. Pandey, S., Olston, C.: User-centric web crawling. In: WWW 2005: Proceedings of the 14th International Conference on World Wide Web, pp. 401–411. ACM, New York (2005)
15. Saporta, G.: Probabilités, analyse des données et statistique. Technip (2006)
16. Sia, K.C., Cho, J., Cho, H.-K.: Efficient monitoring algorithm for fast news alerts. IEEE Trans. on Knowl. and Data Eng. 19(7), 950–961 (2007)
17. Sia, K.C., Cho, J., Hino, K., Chi, Y., Zhu, S., Tseng, B.L.: Monitoring rss feeds based on user browsing pattern. In: Proceedings of the International Conference on Weblogs and Social Media, Boulder Colorado, pp. 161–168 (March 2007)
18. Zimmer, C., Tryfonopoulos, C., Berberich, K., Koubarakis, M., Weikum, G.: Approximate Information Filtering in Peer-to-Peer Networks. In: Bailey, J., Maier, D., Schewe, K.-D., Thalheim, B., Wang, X.S. (eds.) WISE 2008. LNCS, vol. 5175, pp. 6–19. Springer, Heidelberg (2008)
19. Zimmer, C., Tryfonopoulos, C., Berberich, K., Weikum, G., Koubarakis, M.: Node behavior prediction for large-scale approximate information filtering. In: 1st International Workshop on Large Scale Distributed Systems for Information Retrieval, LSDS-IR 2007 (2007)